User Hardware Handbook–Computer
C.P.U. BASIC MULTIPLEXER CHANNEL

The information presented herein is, to the best of our knowledge, true and accurate. No warranty or guarantee, expressed or implied, is made regarding the accuracy of information supplied or capacity, performance or suitability of any product or service since the manner of use is beyond our control.

You are advised that you should ensure that the information contained herein has not been superseded.

All our products, materials and services are sold subject to our Conditions of Sale, available on request.

GEC COMPUTERS LIMITED
Elstree Way, Borehamwood, Hertfordshire.
Telephone No. 01-953-2030

Holding Company – the General Electric Company Limited of England

December, 1977

BASIC MULTIPLEXER CHANNEL

CONTENTS

Page

IN-TEXT FIGURES

SUPPLEMENTS

1 INTRODUCTION

The Basic Multiplexer Channel (BMC) performs three Input/Output (I/O) functions within the 4000 system.

(1) It executes all program transfers, over the Normal Interface, at the instigation of the Central Processor Unit (CPU) via the Command Interface.

(2) It controls the transfer, over the Normal Interface, of data between peripherals and main storage;

(3) As a secondary function the BMC executes the reading and writing into main storage on behalf of the Control and Monitor Unit (CMU)

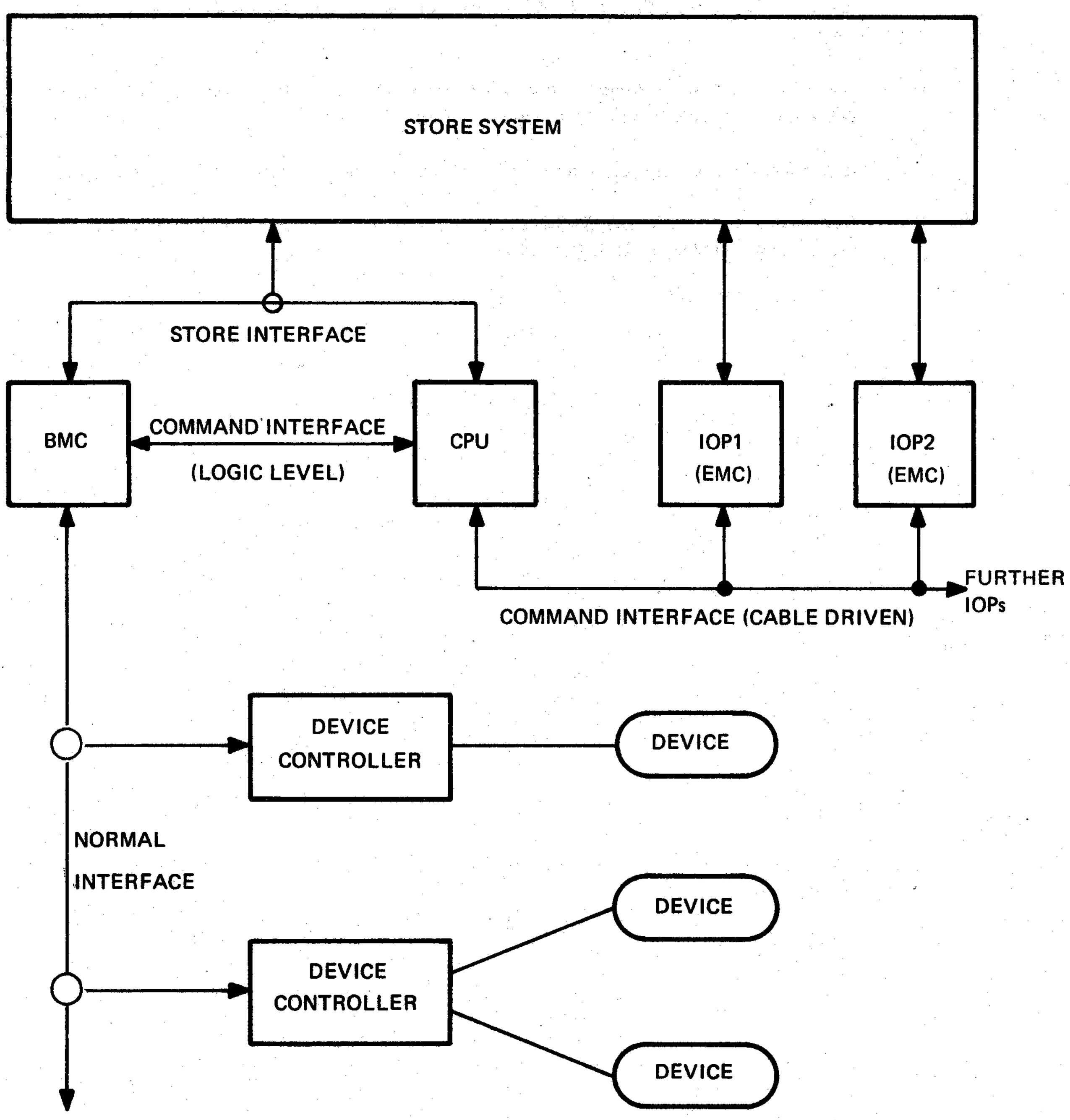


Figure 1: BASIC SYSTEM CONFIGURATION

2 CONFIGURATION

2.1 HARDWARE

The BMC comprises three logic boards which are mounted in the same unit as the processor (CPU) boards. Parts of the logic circuitry, notably the store interface and timing logic, are shared between the BMC and the CPU.

The BMC generates a logic level version of the Normal Interface which is distributed, via the tracked backplane, to nine board positions (slots). The quantity of slots is reduced by one for each main store unit connected to the system. Alternatively, the 4070 processor may be fitted with internal store, which occupies four slots.The remaining slots can be used to house peripheral controllers or to extend the Normal Interface to extension units via Normal Interface link boards. Specialised boards may also be housed in these slots, when necessary, to convert the Normal Interface to suit specific peripherals or equipment.

2.2 SYSTEM

Figure 1 illustrates the manner in which the BMC is interconnected with a 4000 system.

The BMC transfers data to and from main storage via the Store Interface, which it shares with CPU, the BMC havingpriority in the event of simultaneous requests from BMC and CPU. Data transference between the BMC and a peripheral is made via the Normal Interface, whilst the overall control of the Input/Output facilities by CPU is exercised by means of the Command Interface.

A maximum of eight Input/Output Processors (IOP) may be connected to a 4000 system. The BMC is the basic IOP version and as such is connected to Channel O (lowest priority) of the Command Interface. The BMC uses a special backplane version of the Command Interface, all other IOP's using a cable driven version and an independent store interface.

2.3 NORMAL INTERFACE

Peripheral devices are connected to the BMC via the 16 bit version of the Normal Interface as described in the Interface Manual.

The primary interface is provided with 16 READY lines. Thus a maximum of 16 autonomous peripheral controllers may be connected directly to the BMC. However additional READY lines can be made by use of an extension unit which has READY multiplexing facilities, allowing extra controllers tobe connected to the BMC.

The BMC can handle up to 256 concurrent autonomous transfers on up to 256 ways. A transfer occuring on a specific WAY is controlled by one of 256 Way Control Blocks (WCB) which are held in main store and are accessible to the BMC. Each WCB contains information relating to the position and extent of the buffer area of store to be used by the transfer. The WCB is employed to report status and error information to the CPU at the completion of a transfer.

The BMC is also able to perform Program Transfers, initiated by the CPU, using a 16 bit version of the Normal Interface (8-bit Way No. + 8 bit sub-address).

3 PRINCIPLES OF OPERATION

The BMC performs two types of transfer, Program Transfers and Autonomous Transfers, each of which is described in the following paragraphs.

3.1 PROGRAM TRANSFERS

During a Program Transfer a single byte or halfword of data is transferred between the CPU registers and a peripheral. The cycle is initiated by the CPU via the Command Interface, but the data is passed to and from the BMC via a block of fixed locations in store known as the I/O Command Block (IOCB). The IOCB also informs the BMC of the Way No. and sub-address of the peripheral. Program transfers are used to output control information to a given peripheral and to input/output data for peripherals that do not use the Autonomous Transfer facilities.

3.2 AUTONOMOUS TRANSFERS

Autonomous Transfers are performed as a sequence of Autonomous Interface cycles between a peripheral controller and main store. The transfers are originated by the peripheral controller under the supervision of the BMC and the Way No. is written into the IOCB. A cycle is requested by a peripheral controller by the raising of its READY line whereupon the BMC allocates Autonomous Interface cycles on a fixed priority basis.

(a) *Autonomous Data Cycles*

During an autonomous data cycle one or more bytes or halfwords of data are transferred between a peripheral controller and main store via a specific WAY. The Way Control Block (WCB) for that WAY defines the location in main store involved in the transfer. A peripheral controller can request repeated Data Cycles, single or burst mode, until it has no more data to transfer or until the buffer area, in main store, for the WAY is exhausted.

(b) *Autonomous Status Break Cycles*

During an autonomous status break cycle a byte of status information is sent by a peripheral controller to the BMC. This information is written into the WCB for that particular WAY by the BMC. The BMC then informs the CPU, by sending an interrupt along the Command Interface, that a status break has occurred for the peripheral. Status breaks are normally generated to inform the CPU of the termination of an Autonomous Data Transfer.

3.3 INTERFACE CYCLE PRIORITY

The BMC allocates interface cycles on a fixed priority basis. Direct program transfers are given priority over all other service requests. Autonomous transfer requests are serviced on a fixed priority basis, the priority depending on the READY line to which the requesting peripheral is connected. READY 0 has the highest priority and READY 15 the lowest.

4 WAY CONTROL BLOCKS

Autonomous transfers over the Normal Interface of the BMC are controlled by Way Control Blocks (WCB) one for each Way No. A maximum of 256 WCB's are available to the BMC in main storage, starting at an absolute address of 128. The number of Ways, and hence WCB's available to the BMC is set by a pre-patched 4-bit number known as the Way Range (R). This range R is changeable by a switch within the BMC, where R controls the number of WAYS available as follows:

R	Max Way No.
0	15
1	31
etc.up to	
15	255

The address of the first byte of the WCB for WAY 'N' is given by :-

Absolute Way Address = 128 + 8N

4.1 FORMAT OF A WCB

Each WCB has the following format:-

ENTRY 8		PROCESS 8
DEV STATUS 8		ERROR STATUS 8
XADDR 2	COUNT 14	
ADDRESS 16		

4.2 WCB FIELDS

ENTRY AND PROCESS

ENTRY and PROCESS together define the Process to be initiated when an Status Break occurs on the WAY in question. This halfword of the WCB is only used by NUCLEUS and not by the BMC.

DEVICE STATUS

DEVICE STATUS is used to transmit status information concerning transfers, along a specific WAY, to the CPU. During a Status Break cycle a peripheral controller sends a byte of status information to the BMC, the BMC subsequently writes this information into DEVICE STATUS. If any time the BMC detects an error condition as a result of a transfer over the WAY, it inserts a byte of error status information into ERROR STATUS. The BMC will not however, interrupt the CPU if such an error condition occurs.

XADDR and ADDRESS

XADDR and ADDRESS form an 18-bit absolute store address (ABSAD), which is the address of the last byte of the buffer area in store. XADDR forms the m.s. 2-bits of ABSAD. The buffer will normally be used as the source or destination of a string of bytes during an autonomous transfer.

COUNT

COUNT is a 14-bit number which specifies the extent of the buffer area in store. The first byte of the buffer has the address. ABSAD - Count + 1, As the autonomous transfer proceeds COUNT is decremented at each data cycle, such that it always indicatesthe number of bytes still to be transferred.

5 INPUT/OUTPUT CONTROL BLOCK

5.1 GENERAL

The Input/Output Control Block (IOCB) is a fixed area of store, comprising 20 bytes and starting at absolute address 48.

The format of IOCB is as follows:-

48	DEVICE ADDRESS
50	DATA
52	INTADDR 0
64	
66	INTADDR 7

The IOCB fields are described in 5.2.

5.2 IOCB FIELDS

DEVICE ADDRESS

DEVICE ADDRESS is used by the CPU to define the address of a device, on an IOP, upon which a CPU initiated operation is to be performed. In the case of the BMC, DEVICE ADDRESS contains the WAY No. N

DATA

DATA is employed to contain a halfword of information transferred to or from a peripheral and main storage during a CPU initiated operation.

INTADDR

The eight INTADDR locations are used by IOP's when Interrupts are signalled to the CPU. The BMC is allocated INTADDR 0 and, during a Status Break cycle, will write into it the address of the WCB corresponding to the WAY upon which the Interrupt occurred.

SIGNAL

On completion of a Command Interface Cycle involving the CPU and the BMC the BMC will overwrite a 4-bit number into DEVICE ADDRESS to indicate the success or otherwise of the transfer. These 4-bits are known as the SIGNAL bits in this context. Further details may be found in section 8 of this manual.

NOTE Further details on the function of the Command Interface may be found in the Interface Manual.

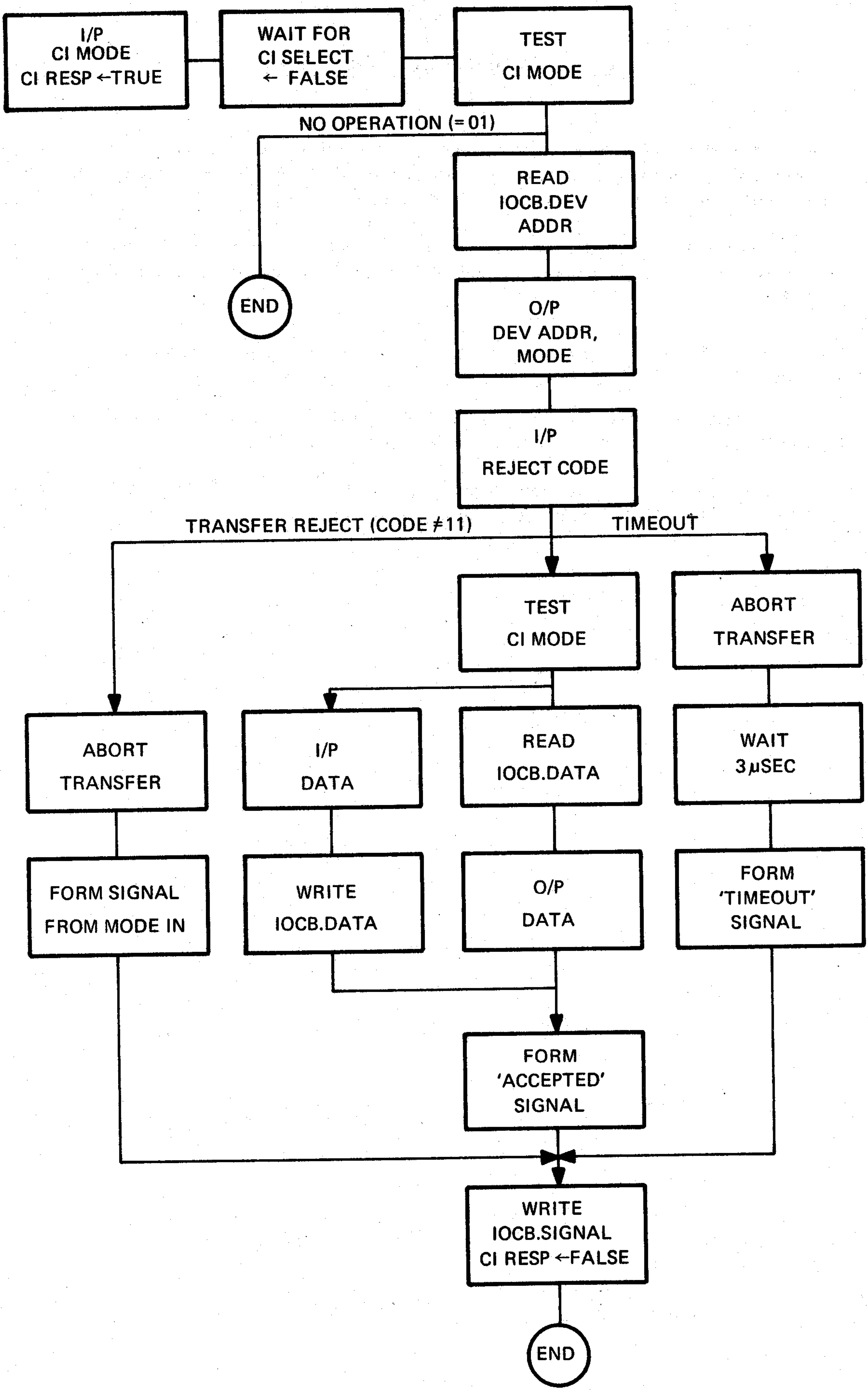


Figure 2: PROGRAM TRANSFER CYCLE

6 PROGRAM TRANSFER OPERATION

The BMC performs Program Transfer cycles via the Normal Interface at the instigation of the CPU. During program transfer operations the BMC employs two locations in the IOCB which contain the WAY No. for the required transfer and a halfword to or from which data is to be transferred.

The CPU having loaded the required data into the IOCB, then causes the BMC to perform a Program Transfer by means of a Command Interface operation. A Summary of Command Interface operation is given in Supplement 1.

The BMC then proceeds to perform Transfer cycle on the Normal Interface as soon as any cycle currently in progress has been completed.

If the device fails to reply within 30/us a time-out error occurs and the BMC will disengage (para 8-1) whilst a rejection causes the BMC to terminate the Normal Interface cycle (para 8-2)

At the completion of a program transfer cycle the BMC writes four SIGNAL bits into the DEVICE ADDRESS location of the IOCB. (The permitted code combinations of the 4 bit SIGNAL are given in para 8.5).

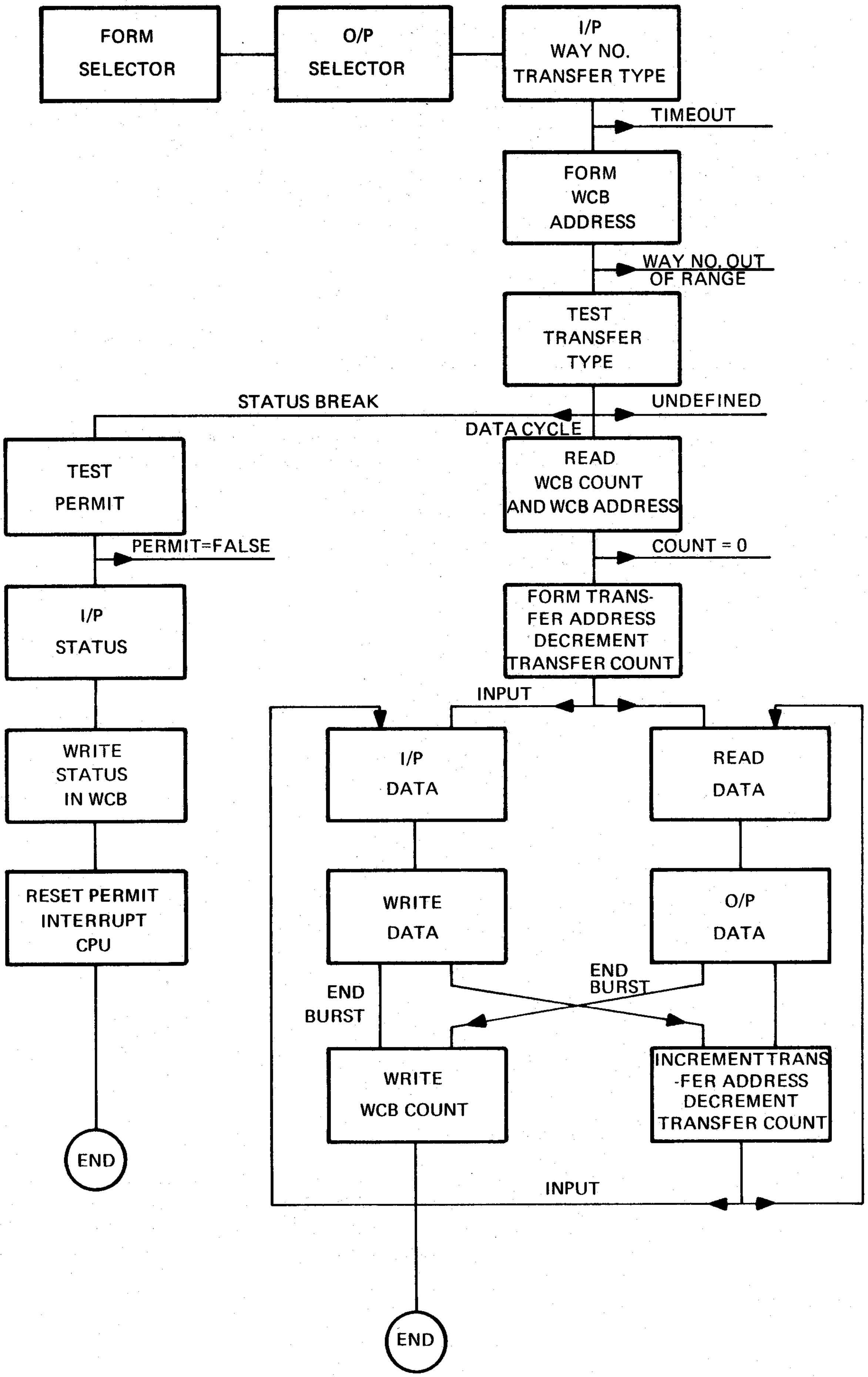


Figure 3: AUTONOMOUS TRANSFER CYCLE

7 AUTONOMOUS OPERATION

An Autonomous peripheral may at any time request data and status transfers by raising its READY line. In an autonomous data cycle, data is transferred between the peripheral device and an area of store,defined by the WCB, whilst in an autonomous status break cycle the BMC transfers status information from the peripheral into the WCB and at the same time sets Interrupt true.

The flow chart for both these cycles is shown in Fig.3 and a summary of Normal Interface operations is given in Fig. 4.

7.1 BASIC OPERATION

(a) When the BMC detects that one or more READY lines are true, it generates a 4-bit READY NUMBER which denotes the highest priority READY.

(b) The BMC now selects the appropriate peripheral controller by means of a SELECT sub-cycle on the Normal Interface. This consists of broadcasting the READY NUMBER on the INFO-OUT lines, setting MODE-OUT = 00 (for autonomous transfers) and setting OUTTIME true.

(c) The peripheral controller on recognising its own READY NUMBER, performs the RESPONSE sub-cycle by placing a 2-bit code, defining the type of autonomous transfer required, on the MODE-IN lines, the Way No. on the INFO-IN lines and by making ENGAGED true.

The 2-bit MODE-IN code is as follows:-

MODE-IN		CYCLE TYPE
0	1	
0	0	NOT USED
0	1	Autonomous Status Break
1	0	Autonomous Data Output
1	1	Autonomous Data Input

If an Autonomous Data Cycle is requested, the INFO-IN MO is used as a width marker to define whether byte or halfword transfers are required.

INFO-IN MO	Width
0	Single byte transfer
1	Halfword (2 byte) transfer

If the peripheral controller fails to respond within 30µs a Time-out Error occurs (see para 9.1)

(d) The BMC now forms the Way Address (Way address = 128 + 8 x Way No.) and checks that the Way No. (N) is within that defined by the Way Range (R). If N > R an error trap is generated (see para 9.2) The type of transfer defined by the MODE-IN bits is now executed, the illegal combination of MODE-IN = 00 causes error action to be taken (para 9.3).

7.2 AUTONOMOUS DATA CYCLE

(a) The Address and Count locations of the WCB are read and an 18-bit address ABSAD and a 14-bit TFER COUNT are formed. If however COUNT = 0 an error is reported and the transfer is aborted (para 9.4).

(b) TFER COUNT is then decremented by 1 or 2 dependent upon the value of the Width Marker:-

WIDTH MARKER	EFFECT
0	TFER COUNT = TFER COUNT - 1
1	TFER COUNT = TFER COUNT - 2

The BMC then forms the transfer address from ABSAD and TFER COUNT. This transfer address is the address of the first byte or halfword of data transferred in the cycle and is formed as follows:-

TFER ADDRESS = ABSAD - TFER COUNT

(c) The BMC now performs one or more Data sub-cycles on the interface depending on the type of transfer operation specified by the MODE-IN bits (para 7.1 (3). The first sub-cycle being initiated by making PROCEED true.

(d) Input Sub-Cycle

If DATA INPUT is requested the peripheral controller places a byte or halfword of data on the INFO-IN lines, together with a 2-bit signal on the MODE-IN lines, and also makes INTIME true. The 2-bit MODE-IN signal indicates to the BMC whether more data sub-cycles are to be executed, as described in the Interface Manual.

The BMC completes the data sub-cycle by sending two MODE-OUT bits to the peripheral controller to indicate whether or not it can comply with the MODE-IN request, and, by making OUTTIME false. After OUTTIME is set false by the BMC the controller may remove its data and make INTIME false.

The data is now written into store at the address specified by TFERADDR, the least significant bit of this address being ignored if halfwords are in use. If the cycle is to be terminated a Cycle Termination sequence is performed (para 7.2 (f)). However if the cyle is to be continued TFER ADDR is incremented by 1 or 2 by decrementing TFER COUNT by 1 or 2 as instructed by the Width Marker and the burst is continued by making OUTTIME true and repeating the input sub-cycle.

(e) Output Sub-Cycle.

When DATA OUTPUT is requested the peripheral controller sends a 2-bit signal on the MODE-IN lines as for the Input Sub-Cycle making INTIME true. The BMC then reads the byte or halfword of data from store defined by TFER ADDR and places this data on the INFO-OUT lines, together with a 2-bit indication on the MODE-OUT lines and also makes OUTTIME false.

After OUTTIME has gone false the peripheral controller may remove the data from the interface and make INTIME false. If the cycle is to be terminated a Cycle Termination sequence is performed (para 7.2 (f)). However if the cycle is to be continued TFER ADDR is incremented and TFER COUNT is decremented as before and the burst is continued by making OUTTIME true and repeating the output sub-cycle.

(f) Cycle Termination

When the BMC wishes to terminate a data cycle, it disengages the peripheral from the interface by making PROCEED false and writes TFER COUNT back into the Count Location of the WCB.

7.3 STATUS BREAK CYCLES

When an autonomous status break cycle is requested by the MODE-IN bits the BMC proceeds as follows:-

(a) The BMC checks its PERMIT lines to determine whether or not there is an outstanding interrupt awaiting action by the CPU. If PERMIT is false an error is signalled and the BMC aborts the transfer (see para 9.5).

(b) If PERMIT is true the BMC initiates a data sub-cycle on the interface by making PROCEED true, the peripheral controllers respond by placing a byte of status information on the INFO-IN lines, setting MODE-IN to 00 and making INTIME true.

((c) The BMC now writes the status information into the DEV STATUS of the WCB. On receipt of INTIME the BMC will set MODE-OUT to 00 and make OUTTIME false. The peripheral controller responds by making INTIME and ENGAGED false.

(d) The BMC now writes the Way Address of the Way concerned into the INADDR 0 location of the Input Output Control Block (IOCB) and raises its interrupt line on the Command Interface. At the same time PERMIT and PROCEED are made false to terminate the Status Break Cycle.

If PERMIT is made false in this step the BMC will not service further READY for 2μs but will perform direct program transfers during this period.

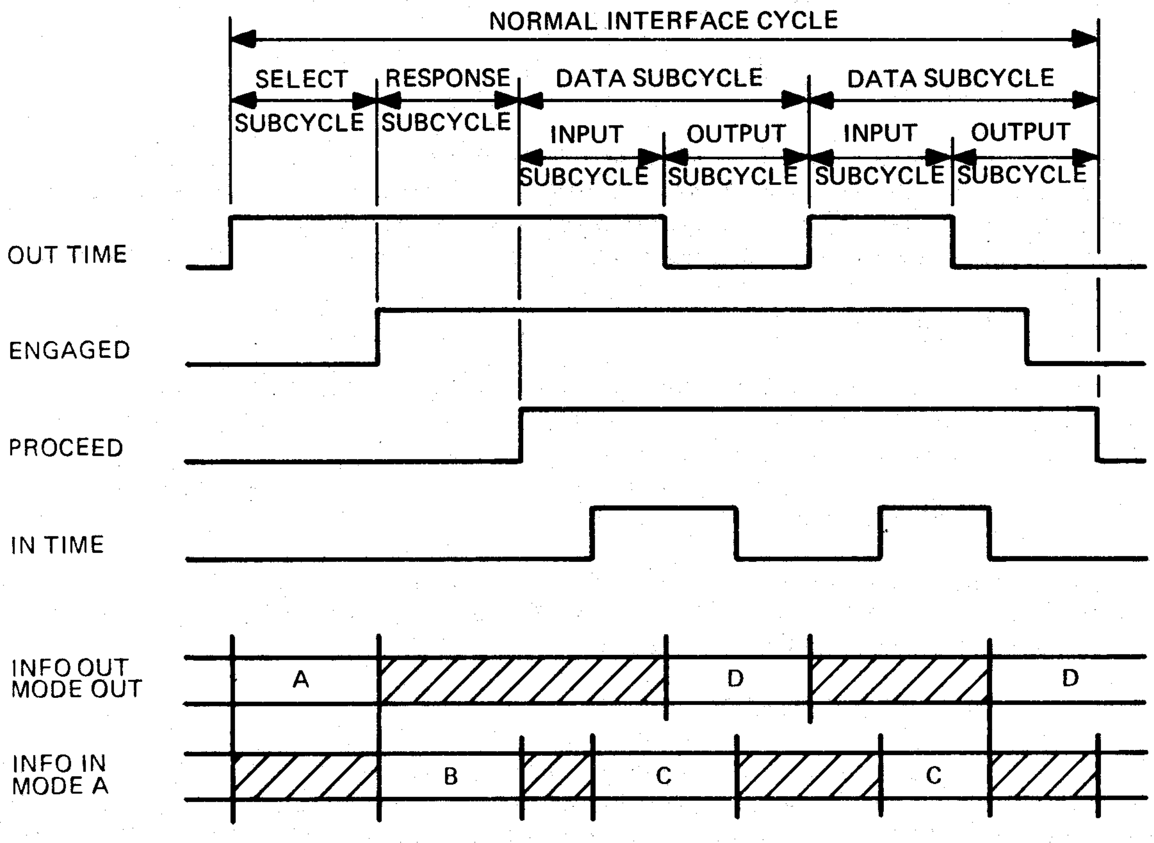
7.4 INTERACTION WITH CPU

NUCLEUS operations initiated by a CALL I/O instruction are interlocked, by means of the Command Interface, with autonomous operations performed by the BMC. The effect of this interlocking is as follows:-

(a) Any such instruction addressing the BMC is not implemented until any Normal Interface cycle in progress has been completed.

(b) If this instruction causes any WCB locations to be read or altered (i.e. read status or load WCB operations) further autonomous cycles cannot occur until the instruction has been completed. Hence there is no possibility of the WCB contents being used by either CPU or BMC whilst being altered by the other.

NUCLEUS Interrupt operations and the AINT instruction in Basic Test mode are also interlocked with the BMC by means of the Command Interface. As stated in 7.3. the PERMIT signal is set false when a Status Break occurs. This signal is returned to true after the Status Location has been read and cleared by the CPU.



TYPE OF CYCLE		SIGNAL	A	B	C	D
AUTONOMOUS	DATA IN	INFO MODE	SELECTOR 0 0	WAY NO. 1 1	DATA BURST CODE	– BURST CODE
	DATA OUT	INFO MODE	SELECTOR 0 0	WAY NO. 1 0	– BURST CODE	DATA BURST CODE
	STATUS 8K	INFO MODE	SELECTOR 0 0	WAY NO. 0 1	STATUS 1 0	– 0 0
PROGRAM	DATA IN	INFO MODE	DEV ADDR 1 0	– RESPONSE CODE	DATA 1 0	– 0 0
	DATA OUT	INFO MODE	DEV ADDR 1 1	RESPONSE CODE	– 1 0	DATA 0 0

Figure 4: NORMAL INTERFACE SYSTEM SUMMARY

8 ERRORS DURING PROGRAM CYCLES

8.1 TIMEOUT

If the peripheral controller fails to reply with ENGAGED within 30μs of the BMC making OUTTIME true then a Timeout error will occur, the BMC will automatically disengage the peripheral from the interface, by making OUTTIME false, and will wait a further 3μs before servicing any further transfer requests. The BMC writes the appropriate SIGNAL bits into the IOCB and terminates the Command Interface cycle.

8.2 DEVICE REJECTION

The BMC will terminate the Normal Interface cycle, by making OUTTIME false, any time that it detects that the MODE-IN bits indicate rejection of a transfer. An appropriate SIGNAL code is written into the IOCB and the Command Interface cycle terminated.

8.3 STORE FAILURE

The occurrance of either a store parity or timeout error, during a direct program cycle will cause the BMC to abort the transfer by making OUTTIME and PROCEED false. The appropriate signal bits are then written into the IOCB and CI RESPONSE is made false.

8.4 OVERALL TIMEOUT

If during direct program cycle either INTIME or ENGAGED are still true 75μs after PROCEED has gone true an overall timeout will occur, the BMC will abort the transfer in progress by making PROCEED, and if necessary OUTTIME, false and not service further request for a period of 3μs. The BMC then writes the appropriate signal bits into the IOCB and terminates the Command Interface cycle.

8.5 SIGNAL BITS

The SIGNAL BITS are coded as follows:-

	SIGNAL			MEANING	SET IF
N 12	Z 13	OF 14	CA 15		
0	0	0	0	Transfer accepted	MODE–IN code 11
0	0	1	1	Unspecified device rejection	MODE–IN code 00
0	1	1	1	Invalid function	MODE–IN code 01
1	0	1	1	Device busy (or PC BUSY)	MODE–IN code 10
0	0	1	0	Device Timeout	No response
1	1	1	0	Store Parity failure	Parity fail at any store read
1	1	1	1	Store timeout	Timeout of any store access
0	1	1	0	Overall timeout	

Bits 0 - 11, the DEVICE ADDRESS location in the IOCB,are zeroed by the BMC when it writes away the SIGNAL bits.

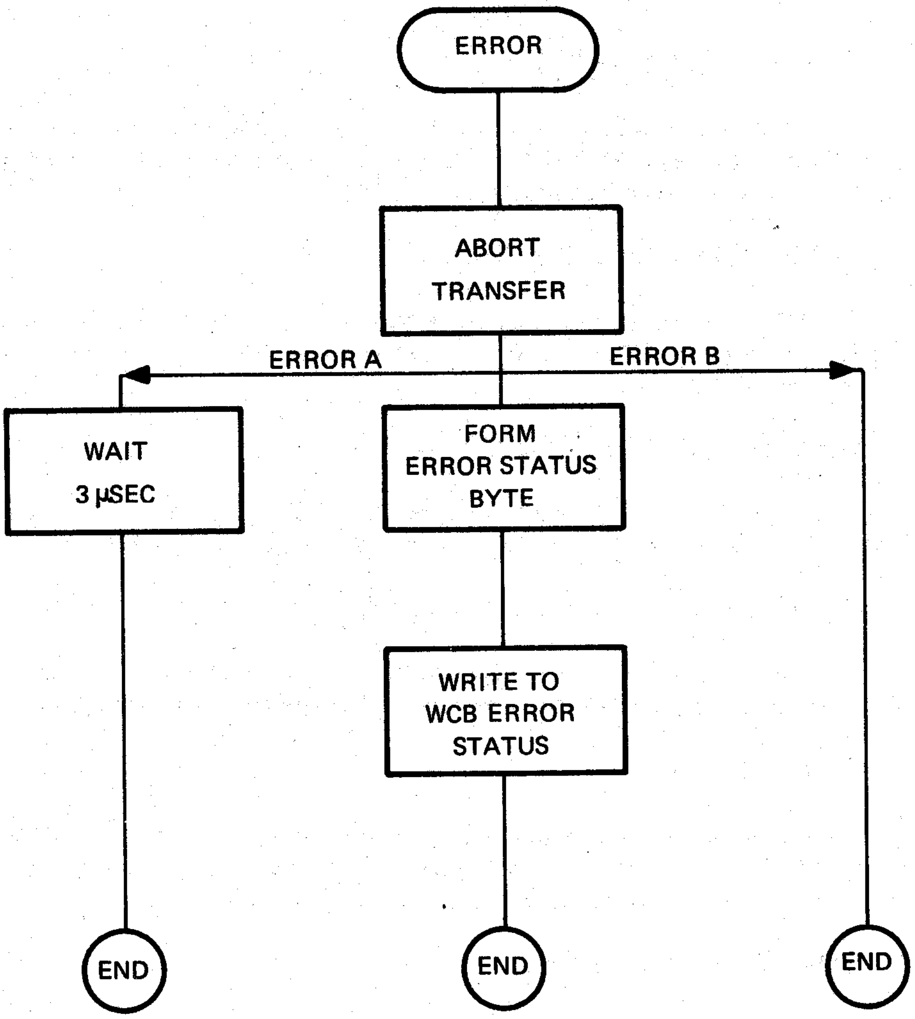


Figure 5: ERROR PROCEDURE—AUTONOMOUS TRANSFER

9 ERRORS DURING AUTONOMOUS TRANSFERS

The following errors are detected by the BMC during autonomoustransfers, error procedure is shown in figure 5.

9.1 TIMEOUT

If during the SELECT sub-cycle no reply has been received from a peripheral controller within 30 µs of OUTTIME going true, a timeout error will occur. The BMC will disengage the interface by making OUTTIME false and will not service requests for a further 3 µs.

9.2 WAY NO OUT OF RANGE

If the BMC receives, during the RESPONSE sub-cycle, a Way No. greater than that permitted the BMC aborts the transfer by making OUTTIME false. After the peripheral has disengaged by making ENGAGED false the BMC is free to start the next service cycle.

9.3 UNDEFINED OPERATION REQUEST

If the MODE-IN bits received in the RESPONSE sub-cycle specify an undefined operation the BMC aborts the transfer by making OUTTIME false. An error status byte is formed and written into the ERROR STATUS' location of the WCB.

9.4 DATA REQUEST WITH COUNT = 0

If the MODE-IN bits received during the RESPONSE sub-cycle specify a data transfer, but the COUNT location in the WCB is zero, the device is in error and the transfer is aborted (para 9.3).

9.5 STATUS BREAK REQUEST WITH PERMIT FALSE

If the MODE-IN bits received during the RESPONSE sub-cycle specify a status break but an interrupt is still pending (PERMIT false) the device is in error and the BMC aborts the transfer (para 9.3)

9.6 OVERALL TIMEOUT

A timeout error will occur, during an autonomous transfer, any time that ENGAGED or INTIME are true 75 µs after PROCEED has gone true. The BMC will abort the cycle (para 8.3) and report the error (para 9.3). This sequence is employed to guard against excessively long transfer cycles.

9.7 STORE PARITY FAILURE

If during an autonomous transfer, a store parity failure occurs whilst data is being read from store the BMC aborts the transfer (para 9.6) and reports the error (para 9.3).

9.8 STORE TIMEOUT

The action taken for a store timeout occurring during an autonomous transfer is the same as for a store parity failure (para 9.7).

9.9 ERROR STATUS BYTE

The ERROR STATUS bytes for the aforementioned failures are coded as follows:-

BMC

ERROR STATUS			FAILURE
13	14	15	
0	0	0	No error
0	0	1	Undefined operation requested by peripheral
0	1	0	Data request COUNT = 0
0	1	1	Status break request with PERMIT false
1	0	0	Overall timeout
1	0	1	Store parity failure
1	1	0	Store timeout

10 CMU FUNCTIONS

A secondary function of the BMC is the control of the reading and writing, into main storage via, the control and Monitor Unit (CMU); the CMU is described in its own handbook. Five modes of operation are available and are as follows:-

10.1 ENTER ADDRESS

An 18-bit halfword address may be loaded, into the BMC store address register, from the keys on the CMU front panel. This address, which must be in the range 0 to 262143 is used as the start address for a read or write function.

10.2 READ MODE

Two functions are available in this mode, read from one address in store and read increment from store. In the first a halfword of data is read from the address entered from the CMU and loaded into the BMC store data register, this data may be displayed by the CMU. In the second function, read increment, data is read from the start address and loaded by the BMC as before but the store address register is incremented by 2 in preparation for the next store access.

10.3 WRITE MODE

Similar facilities are available for write as for read, the difference being that data is read from the CMU keys and written, via the BMC into main storage. In this mode the address is incremented after each store access.

11 INTERFACE CONTROLS

The BMC is reset to a quiescent state by a signal from the CPU, derived from the following conditions:-

(1) INTERLOCK

(2) RAILS CORRECT AND MAINS CORRECT

(3) RESET KEY

Details of these conditions and their effect on the BMC are given in the following paragraphs.

11.1 INTERLOCK

INTERLOCK is used to warn the BMC of an impending power failure in critical sections of a 4000 system e.g. a store module. During normal operation INTERLOCK is held true; but in the event that it goes false the BMC will be prevented from servicing any further requests after the completion of an existing transfer. The BMC will remain in the quiescent RESET state until such time as INTERLOCK goes true again.

11.2 RAILS CORRECT (RC) AND MAINS CORRECT (MC)

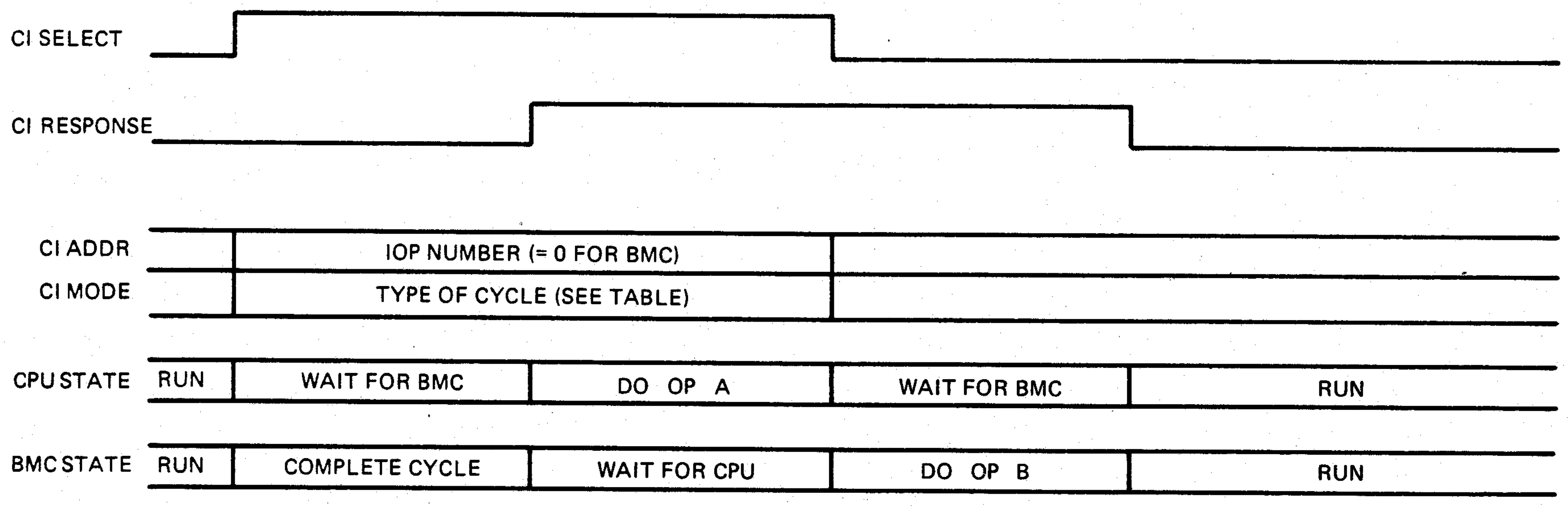
RC and MC are used by the BMC in the same way as INTERLOCK. These two signals provide information to the BMC that when they are both true correct power is applied to the BMC. At least 1ms is given prior to power collapse, by these signals, for the BMC to complete any current transfers and for it to go into the quiescent RESET state.

11.3 RESET KEY

This facility enables the BMC to be reset by depressing the RESET key on the CMU; in this instance IO RESET is momentarily false and then made true again.

12 NORMAL INTERFACE PARAMETERS

16-bit Program Transfer Limit -		1 - 16-bit halfword
Addressing Width		16-bits
Autonomous Transfer Facilities		8-bits and 16-bits
Status Word Width		8-bits
Maximum number of Ways for Autonomous Data transfers		256
Maximum number of Ways for Autonomous Status Breaks		256
Number of READY lines		16
Timeout Limits:-	Device timeout	OUTTIME↑ to ENGAGED↑ approximate 30μs
	Overall Timeout	OUTTIME↑ to ENGAGED↓ approximate 75μs



CI MODE	TYPE OF CYCLE	DO OP A	DO OP B	NOTE
0 0	Reset Interrupt	–	PERMIT := TRUE	BMC DOES NOT STOP OPERATION
0 1	Wait for CPU	Load a WCB	–	DROP CI RESPONSE AS SOON AS CI GOES FALSE
1 0	Direct Input	–	NI Input Cycle	WRITE SIGNAL BITS TO IOCB BEFORE DROPPING CI RESPONSE
1 1	Direct Output	May Load a WCB	NI Output Cycle	

Figure 6: COMMAND INTERFACE CYCLE SUMMARY

Supplement 1: OUTLINE OF COMMAND INTERFACE OPERATION

When the CPU performs a cycle on the Command Interface, it outputs two CI MODE bits and then makes CI SELECT 'true'. The CI mode bits define the type of operation required of the BMC as follows:-

MODE 0	 1	REQUIRED OPERATION
0	0	Reset Interrupt
0	1	Wait for CPU
1	0	Program Input
1	1	Program Output

The action taken by the BMC for each different cycle is shown in Figure 3.

The BMC does not respond with the exception of CI MODE = 00. until the completion of any existing autonomous transfer cycle and has reached a quiescent state. Subsequent action by the BMC is as follows:-

(1) WAIT FOR CPU

If WAIT FOR CPU is requested the BMC completes any existing interface operation replies with CI RESPONSE true and then waits until the CPU responds with CI SELECT false.
This mode is used by NUCLEUS during the CALL (I/O) instruction (see NUCLEUS MANUAL) to delay autonomous action by the BMC when the CPU loads a Way Control Block and when it reads the Status byte from the WCB.

(2) RESET INTERRUPT

For RESET INTERRUPT CI RESPONSE is made true immediately and the BMC drives its PERMIT line false, without effecting the normal operation of the BMC.

This action permits the BMC to service further Status Breaks when required. This mode is used by NUCLEUS, during Interrupt handling, to reset the incoming Interrupt (see NUCLEUS Manual). The instruction AINT provides the same facility in the Basic Test Mode (see 4000 Instruction Manual).

(3) PROGRAM TRANSFER

If either PROGRAM INPUT or PROGRAM OUTPUT is requested the BMC responds, after completion of any existing interface operation by making CI RESPONSE true. The CPU then makes CI SELECT false, at which point the required Normal Interface cycle is performed. The Command Interface cycle is then terminated by making CI RESPONSE false.